palro

Developer Documentation

アプリケーション開発ガイド

(SPC/PAPPS 共通)

© 2012 FUJISOFT INCORPORATED. All rights reserved.

palro

目　次

1. はじめに 2

1.1 本書対象者 2
1.2 PALRO システムソフトウェアのバージョン 2
1.3 事前準備 2

2. なにを作ろう？ 3

3. アプリケーションに必要なファイルを用意しよう 4

3.1 アプリケーションに必要なファイル 4

4. プログラミングしよう 6

4.1 ヘッダファイルを作成する 6
4.2 ソースファイルを作成する 7
4.3 makefile を作成する 9
4.4 プログラムファイルを作成する 10

5. PALRO にインストールしよう 11

5.1 インストールのための準備 11
5.2 インストールする 12
5.3 テストする 13
5.4 アンインストールする 13

6. PALRO Garden に公開しよう 14

7. アプリケーションを拡張しよう 15

7.1 アプリケーションに個別情報を追加したい 15
7.2 プライベート会話ファイルを使用したい 19

1. はじめに

本書は、Sapie Program Component(以降、SPC)、および PALRO Applications(以降、PAPPS)の開発ならびにインストールについて説明します。
また、本書ではサンプルを使用して説明しますが、明記しない限り SPC を対象としています。PAPPS については、別冊 *"PAPI(PALRO Application Programming Interface) リファレンス"*、*"SAPI(Sapie Application Programming Interface) リファレンス"* ならびに、開発環境【 アプリケーション開発用 VMware イメージ 】に含まれるサンプルソースを参照してください。

	SPC (Sapie Program Component)	PAPPS (PALRO Applications)
特長	話す・聞く・歩くなどの命令をシーケンシャルに記述するシンプルなインターフェースで、簡単に機能を追加可能	プリミティブな API(PAPI/SAPI)で、独自に PALRO を制御するインターフェースが提供され、複雑な機能を追加可能
難易度	やさしい	難しい
バージョン差異による互換性	高い	低い

1.1 本書対象者

本書は PALRO の使用方法について既に知識のある方を前提としています。
加えて、以下の項目の有識者を対象としています。

- プログラミング言語　C++
- Ubuntu Linux 上でのソフトウェア開発経験

1.2 PALRO システムソフトウェアのバージョン

本書の内容は、PALRO システムソフトウェアバージョン 3.2 を対象としています。

1.3 事前準備

(1) 開発環境の構築

詳細は、PALRO Garden の【 デベロッパー 】-【 ドキュメント 】内の *"アプリケーション開発環境構築ガイド"* を参照してください。

アプリケーション開発用 VMware イメージ には、PAPPS のサンプルソースが含まれています。PAPPS 開発の際は、参考にしてください。
インストール先：**/usr/local/src/papps/**

(2) PALRO Application Studio のインストール

詳細は PALRO Garden の【 デベロッパー 】-【 ドキュメント 】内の *"PALRO Application Studio ガイド"* を参照してください。

2. なにを作ろう？

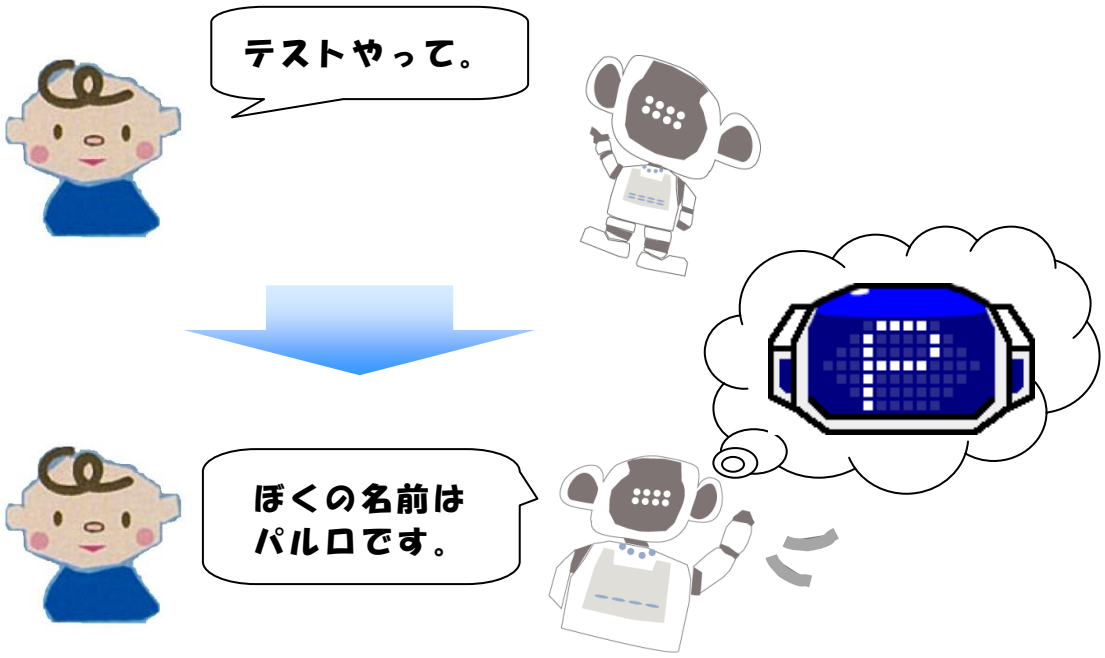
誰をどうやって驚かせましょうか？
どんな風に喜んでもらいましょうか？

あなたにそのアプリケーションのイメージと、ちょっとした Linux と C/C++の知識、それと PALRO さえあれば、今まで誰も体験したことのないような驚きをあなたが提供することができるのです。

アプリケーション(SPC) を開発することで、PALRO に簡単にオリジナルの機能を追加することができます。
人を見たり、手を動かして話すなどの基本的な動作は、すべてシステムソフトウェアが代行するため、開発者は機能だけを考えれば良いことになります。
SPC が提供する機能については、別冊 *〝SPC(Sapie Program Component) リファレンス〟* を参照してください。

本書では、【 SampleComponent 】という名前のサンプルアプリケーションを例に、SPC の作り方を説明します。
SampleComponent は、以下の動きをするアプリケーションです。

(1) PALRO に『 テストやって。 』と話しかけると、
(2) 顔の LED に「 P 」と表示され、
(3) 左腕を上にあげて、
(4)「ぼくの名前はパルロです。」と発話する。



3. アプリケーションに必要なファイルを用意しよう

なにを作るか決まったら、PALRO のアプリケーションを構成するファイルを用意します。

PAPPS についても同様に、必要なファイルを用意します。

3.1 アプリケーションに必要なファイル

PALRO のアプリケーションを構成するファイルを以下に示します。
SampleComponent では、下記の「(1) (2) (3) (4) (5)」を使用します。

(1) プログラムファイル（必須）
アプリケーションの実行ファイルです。作成方法については *〝本書 4. プログラミングしよう〟* を参照してください。

(2) アプリケーションコンフィグファイル（必須）
アプリケーションをインストールする際に必要な登録情報や、アプリケーションの動作に必要な設定値を定義した XML 形式のファイルです。詳細は、別冊 *〝PALRO Application Studio ガイド〟* を参照してください。

(3) トリガーワードファイル（任意）
アプリケーションを起動するためのトリガーワードを定義したテキスト形式のファイルです。アプリケーションの起動を音声で行う場合に必要です。詳細は、別冊 *〝PALRO Application Studio ガイド〟* を参照してください。

SampleComponent 用のアプリケーションコンフィグファイル、およびトリガーワードファイルを作成する場合、PALRO Application Studio の設定項目は以下のように設定してください。

項目	設定内容
アプリケーション名	**SampleComponent** と入力する
カテゴリ	指定なしを選択する ※ カテゴリを入力すると、PALRO がゲームやクイズを提案するときに、開発したアプリケーションが選ばれる可能性があります。

項目	設定内容
SPC/PAPPS	**SPC(Sapie Program Component)**を選択する ※ PAPPS の場合は、【 PAPPS 】を選択
タイトル	**PALRO Sample** と入力する
読み	**パルロサンプル**と入力する
アプリケーションのトリガーワードを作成する	チェックを入れる
始めるときのトリガーワード	**テスト** を **やって/して** と入力する
終わるときのトリガーワード	チェックを入れない

(4) PALRO LED ファイル（任意）
PALRO の顔の LED の表示内容を定義したファイルです。詳細は、別冊 *〝PALRO Application Studio ガイド〟* を参照してください。

SampleComponent で使用する PALRO LED ファイルは、SampleLED.led という名前で作成します。なお、PALRO Garden の【 デベロッパー 】-【 ドキュメント 】-【 アプリケーション開発ガイド 】にある【 本書で使用した PALRO LED ファイルはこちら 】からダウンロードすることもできます。

(5) PALRO モーションファイル（任意）
PALRO の動きを定義したファイルです。バンザイや敬礼など特定の動作や、ダンスなどを PALRO に実行させる際に必要です。詳細は、別冊 *〝PALRO Motion Editor ガイド〟* を参照してください。

SampleComponent で使用する PALRO モーションファイルは、SampleMotion.act という名前で作成します。なお、PALRO Garden の【 デベロッパー 】-【 ドキュメント 】-【 アプリケーション開発ガイド 】にある【 本書で使用した PALRO モーションファイルはこちら 】からダウンロードすることもできます。

(6) プライベート会話データファイル（任意、PAPPS 専用）
アプリケーションが専用で使用する会話データを定義したテキスト形式のファイルです。アプリケーションの実行中に独自の会話を行う PAPPS を作成する場合に必要です。
SPC を開発する場合は、専用の API が用意されていますので、プライベート会話データファイルを作成する必要はありません。作成方法については *〝本書 7.2 プライベート会話ファイルを使用したい〟* を参照してください。

4. プログラミングしよう

アプリケーションに必要なファイルが準備できたら、SampleComponent のプログラムファイルを作成します。

プログラムソースの文字コードは、必ず **UTF-8N(BOM 無し)**を使用してください。これ以外の文字コードでは、正しく動作しない場合があります。

ここで説明していない SPC の内容については、別冊 *"SPC(Sapie Program Component) リファレンス"* を参照してください。

4.1 ヘッダファイルを作成する

ヘッダファイルの内容を以下に記載します。この内容を **SampleComponent.h** というファイル名で保存してください。

```
#ifndef _SAMPLE_COMPONENT_H_
#define _SAMPLE_COMPONENT_H_

// ヘッダファイル……(1)
#include <spc/spcbase.h>
using namespace spc;

// 作成する SPC のクラス……(2)
class CSampleComponent : SPCBase {
  public:
        // コンストラクタ
        CSampleComponent();

        // デストラクタ
        ~CSampleComponent();

        // SPC 実行開始イベント……(3)
        void onInitialize();
};

#endif // _SAMPLE_COMPONENT_H_
```

(1) SPC 作成用のヘッダファイルです。
(2) SPC は、SPCBase クラスを継承して作成します。このクラスには、発話要求、質問要求、歩行要求など、PALRO の基本機能を実行するためのメソッドが定義されています。
(3) onInitialize 関数は、SPC の起動前に一度だけ実行されるコールバック関数です。SPCBase クラスには、この他にも顔認識や動体検知時などに実行されるコールバック関数が定義されています。

4.2 ソースファイルを作成する

ソースファイルの内容を以下に記載します。この内容を **SampleComponent.cpp** というファイル名で保存してください。

```
#include "SampleComponent.h"

// SPC のインスタンスを定義する ····· (1)
CSampleComponent spcApp;

// コンストラクタ
CSampleComponent::CSampleComponent() {
}
// デストラクタ
CSampleComponent::~CSampleComponent() {
}

//SPC 初期化イベント ····· (2)
void CSampleComponent::onInitialize() {

        // 顔の LED 表示 ····· (3)
        startLED("/usr/palro/etc/apl/SampleComponent/SampleLED.led");

        // モーションを実行 ····· (4)
        makeMotion("/usr/palro/etc/apl/SampleComponent/SampleMotion.act");

        // 発話 ····· (5)
        speak("こんにちは、ぼくの名前は、パルロです");

        // SPC の終了 ····· (6)
        exitComponent();
}
```

(1) 作成する CSampleComponent クラスのインスタンスを、「spcApp」という名前で定義してください。これは、SPC ライブラリが開発者の作成した関数を呼び出すための規約です。
(2) SPC の処理は、SPCBase クラスで定義されたコールバック関数をオーバーライドして作成します。
(3) startLED 関数(顔の LED の表示)には、PALRO LED ファイルのパスを指定します。SampleComponent で使用する PALRO LED ファイルは、インストールすると以下のパスに格納されます。詳しくは、*"本書 5. PALRO にインストールしよう"* を参照してください。
インストール先：/usr/palro/etc/apl/SampleComponent/
(4) makeMotion(モーションの実行)には、PALRO モーションファイルのパスを指定します。SampleComponent で使用する PALRO モーションファイルは、「(3)の PALRO LED ファイル」と同じ場所にインストールされます。
(5) speak 関数(発話)には、発話したい文字列を指定してください。
(6) exitComponent 関数は、SPC を終了するための関数です。

exitComponent 関数を呼び出さなくても、システムソフトウェアが自動的に SPC の実行を中止する場合があります。(PALRO が持ち上げられた場合など)

4.3 makefile を作成する

SampleComponent の makefile を以下に記載します。この内容を **makefile** というファイル名で保存してください。

```
OUTPUT_DIR=.

MOD_NAME=SampleComponent
MOD_SRC=SampleComponent.cpp

MOD_LIB=$(OUTPUT_DIR)/$(MOD_NAME)
MOD_OBJ=$(MOD_SRC:.cpp=.o)
DEP=$(MOD_SRC:.cpp=.d)

INC_PATH=-I./

CFLAGS=-Wall -c -Wno-deprecated
LFLAGS=-Wl

LIB_PATH=-L/usr/local/lib/
LIB_FILE=-lspcif -lspc·····(1)

all:release

release:CFLAGS+=-O3
release:$(MOD_LIB)

debug:CFLAGS+=-g
debug:$(MOD_LIB)

$(MOD_LIB):$(MOD_OBJ)
[TAB]$(CXX) $(LFLAGS) -o $@ $(MOD_OBJ) $(LIB_PATH) $(LIB_FILE)·····(2)

.cpp.o:
[TAB]$(CXX) $(CFLAGS) $(INC_PATH) -MMD -MP -MF"$(@:%.o=%.d)" -o"$@" "$<"

clean:
[TAB]rm -f $(MOD_OBJ) $(MOD_LIB) $(DEP)·····(2)

-include $(DEP)
```

(1) SPC の実行に必要なライブラリとして、「libspcif」および、「libspc」を指定します。
(2) 「 **[TAB]** 」には、タブ文字を入力してください。

4.4 プログラムファイルを作成する

作成したヘッダファイル、ソースファイル、makefile を同じディレクトリに格納し、以下のコマンドを実行してください。

```
$ make
```

SampleComponent という名前のプログラムファイルが作成されれば完了です。

5. PALRO にインストールしよう

ここでは、開発したプログラムファイル(SampleComponent)を PALRO にインストールするための手順を説明します。

PAPPS についても同様の手順でインストールおよび実行することができます。

5.1 インストールのための準備

アプリケーションをインストールするために、必要なファイルを準備します。
以下に、*"本書 4. プログラムミングしよう"* で開発した SampleComponent を例にファイル構成を示します。

```
[ アプリケーション名 ]
    ├ SampleComponent
    ├ [ etc ]
    |     ├ SampleComponent.conf.xml
    |     ├ SampleComponent.ukb
    |     ├ SampleLED.led
    |     ├ SampleMotion.act
    |     └ その他のファイル(音楽ファイルなど)
    └ [ src ]
          ├ SampleComponent.cpp
          ├ SampleComponent.h
          └ makfile
```

※ [] はフォルダー

[etc]フォルダーに格納されたファイルは、全て PALRO にインストールされます。アプリケーションで必要なファイルは、全てここに格納してください。

インストール先： **/usr/palro/etc/apl/(プログラムファイル名)/**

プログラムソースを他の開発者またはユーザーに公開する場合は、[src]フォルダーに格納してください。

5.2 インストールする

アプリケーションのインストールは、PALRO Application Studio を使用します。詳細は、別冊 *〝PALRO Application Studio ガイド〟* を参照してください。
PALRO Application Studio を起動したのち、【 インストール 】画面にて必要な項目を入力してください。
【 パッケージの生成 】を実行したのち【 PALRO へのインストール 】をクリックすると、アプリケーションが PALRO にインストールされます。

項目	設定内容
プログラムファイル	**プログラムファイル**を指定する ※ ここでは、SampleComponent を指定
アプリケーションプロパティの保存フォルダー	*〝本書 5.1 インストールのための準備〟* で作成した**[etc]**フォルダーを指定する
ソースファイルの保存フォルダー	*〝本書 5.1 インストールのための準備〟* で作成した**[src]**フォルダーを指定する ※ プログラムソースを公開する場合のみ必要

5.3 テストする

アプリケーションのインストールが完了したら、実際に動作を確認できます。
SampleComponent では、〝 *本書 3.1 アプリケーションに必要なファイル* 〟でトリガーワード「テスト を やって/して」を設定したので、PALRO に『 テストやって 』と話しかければ、SampleComponent が起動します。

また、開発したアプリケーションをテストしたい場合は、PALRO Application Studio でデバッグ用のログが表示できます。詳細は、別冊 〝 *PALRO Application Studio ガイド* 〟を参照してください。

5.4 アンインストールする

インストールしたアプリケーションは、PALRO コントロールパネルを使用してアンインストールします。詳細は、別冊 〝 *取扱説明書【基本操作】* 〟を参照してください。

PALRO コントロールパネルの【 アプリケーション 】をクリックし、一覧の中から削除したいアプリケーションの【 削除 】をクリックしてください。

プログラムファイルを修正して再度インストールする場合、アンインストールする必要はありません。〝 *本書 5.2 インストールする* 〟に従ってインストールしてください。

6. **PALRO Garden に公開しよう**

開発したアプリケーションは PALRO Garden にアップロードすることで、他の開発者またはユーザーに公開することができます。あなたが開発したオリジナルのアプリケーションを、たくさんの人に使ってもらいましょう。

詳細は、PALRO Garden の【 デベロッパー 】-【 ドキュメント 】内の ″ *アプリケーションアップロードガイド* ″ を参照してください。

アップロードするアプリケーションは、事前に PALRO Application Studio を使用して十分に動作確認したのち、アップロードしてください。

7. アプリケーションを拡張しよう

ここでは、開発したアプリケーションを拡張する方法を説明します。

7.1 アプリケーションに個別情報を追加したい

PALRO コントロールパネルを使用すると、アプリケーションの動作パラメータなどを設定することができます。ここでは、PAPPS のサンプルアプリケーションである PALRO Count Sheep を例に、アプリケーションの設定項目を追加する方法を説明します。

PALRO Count Sheep は、アプリケーション開発用 VMware イメージ に含まれています。
インストール先：**/usr/local/src/papps/CountSheep/**

また、PALRO Garden の【 デベロッパー 】-【 ドキュメント 】-【 アプリケーション開発ガイド 】にある【 本書で使用した PALRO Count Sheep はこちら 】からもダウンロードできます。

PALRO Count Sheep は、ユーザーが設定した最大値まで、羊または執事を数える、というアプリケーションです。アプリケーションに設定項目を持たせるには、アプリケーションコンフィグファイルを編集し、設定する項目を記述する必要があります。

PALRO Count Sheep は、以下の二点をユーザーが設定できるようにしています。
・カウントする最大値 （以下の設定例 ： <title>数え上げ数</title>）
・数える対象 （以下の設定例 ： <title>数え上げ対象</title>）

これらの設定項目を記述した、アプリケーションコンフィグファイルの内容は以下の通りです。

設定例の網掛け部分は、PALRO Application Studio で自動的に作成されるため、詳細説明を省略します。

```
<?xml version="1.0" encoding="utf-8"?>
<palro_application name="CountSheepApp" version="3.2.0" releasedate="2012/11/09" type="PAPPS" mode="transient" appkind="other" language="ja">
  <title>PALRO Count Sheep</title>
  <titlespk>カウントシープ</titlespk>
  <description>PAPPSでアプリケーションコンフィグファイルを読み込むサンプルアプリケーションです。</description>
  <section order="1" name="common">                                    …(1)
    <title>動作設定</title>                                            …(2)
    <description></description>                                        …(3)
    <configuration order="1" name="count" display="yes" modify="yes" conftype="text" formtype="text">   …(4)
      <title>数え上げ数</title>                                        …(5)
      <description>幾つまで数え上げるかを設定します。(1～10)</description>   …(6)
      <value>10</value>                                                …(7)
      <defaultvalue>10</defaultvalue>                                  …(8)
    </configuration>
    <configuration order="2" name="kind" display="yes" modify="yes" conftype="text" formtype="text">
      <title>数え上げ対象</title>
      <description>0:ひつじ 1:しつじ</description>                     (9)
      <value>0</value>
      <defaultvalue>0</defaultvalue>
    </configuration>
  </section>
</palro_application>
```

：自動生成された部分

タグ	属性	設定内容
(1)section	設定項目の大分類を定義します。 大分類は複数定義できますが、以下の属性値を重複させることはできません。	
	order	【 コントロールパネル 】-【 アプリケーション 】-【 設定 】で表示される順序を定義します。複数の大分類を定義した場合、ここで指定した値が小さい順に表示されます。 ※半角数字を使用します。
	name	システムソフトウェアが大分類を識別するための識別子を定義します。 ※半角数字を使用し、他の大分類と重複しない識別子を定義します。
(2)title	大分類の名称を定義します。	
(3)description	大分類の説明を定義します。	
(4)configuration	設定項目の小分類を定義します。 小分類は複数定義できますが、以下の属性値を重複させることはできません。	
	order	【 コントロールパネル 】-【 アプリケーション 】-【 設定 】での表示で、大分類の中での表示順序を定義します。複数の小分類を定義した場合、ここで指定した値が小さい順に表示されます。 ※半角数字を使用します。
	name	システムソフトウェアが小分類を識別するための識別子を定義します。 ※半角数字を使用し、他の小分類と重複しない識別子を定義します。
	display	小分類を表示する場合は **yes** を、表示しない場合は **no** を定義します。
	modify	設定項目の変更可否を定義します。現在は **yes**（変更可能）のみ対応しているため、**yes** を定義します。
	conftype	設定項目のタイプを定義します。現在は **text**（テキストボックス）のみ対応しているため、**text** を定義します。
	formattype	入力できるフォーマットを定義します。現在は **text**（テキスト入力）のみ対応しているため、**text** を定義します。
(5)title	小分類の名称を定義します。	
(6)description	小分類の説明文を定義します。	
(7)value	【 コントロールパネル 】-【 アプリケーション 】-【 設定 】で表示される画面で入力された値を保持します。	
(8)defaultvalue	将来拡張用のタグです。定義する必要はありません。	
(9)configuration	「(4) ~ (8)」と同様に、二つ目以降の小分類を定義します。	

編集したアプリケーションコンフィグファイルをインストールし、【 PALRO コントロールパネル 】の【 アプリケーション 】から【 PALRO Count Sheep 】の【 設定 】選択すると、記述した設定項目が表示されます。

また、アプリケーション(SPC)から設定値を取得する場合は、loadConfig 関数を使用します。上記の例での[数え上げ数]を取得するには、以下のように記述します。

```
loadConfig(”common”, ”count”, value);
```

第一引数に大分類名、第二引数に小分類名を指定すると、第三引数に設定値が返ります。詳細は、別冊 ” *SPC(Sapie Program Component) リファレンス* ” を参照してください。

7.2 プライベート会話ファイルを使用したい

プライベート会話データファイルは、トリガーワードにより起動されたアプリケーションが専用で使用する会話のデータを定義するもので、PAPPS を作成する際に使用することができます。例えば、アプリケーションの処理中に、質問をする場面があった場合、プライベート会話データファイルに音声認識させる言葉を定義することで、専用の音声認識を行うことができます。尚、SPC を作成する場合は、専用の API が用意されていますので、プライベート会話データファイルを作成する必要はありません。

7.2.1 プライベート会話データファイル作成時の注意点

作成時は以下の点に注意し、テキストエディターで作成します。

- 漢字コード: EUC_JP
- 改行コード: LF(0x0a)
- 全て全角文字で記載
- 項目間はタブ区切り
- 拡張子:.ukb
- ファイル名: アプリケーション名_private.ukb （例: MyApp_private.ukb）

7.2.2 プライベート会話データファイルで定義する内容

プライベート会話データファイルでは、以下の情報を定義します。

① ユーザー辞書
以下で示す、③ユーザー発話に指定する文の"読み"をカタカナで指定します。全てのユーザー発話について、1 行1文で"読み"を指定してください。また、ひとつの文に、複数の"読み"を指定することができます。

② 会話タイトル
会話のタイトルを定義します。

③ ユーザー発話
ユーザーが PALRO に話しかける言葉です。
※ひとつの会話タイトルの中で、同じ言葉を使用することはできません。

④ 回答テキスト
上記③で指定したユーザー発話を音声認識した際にアプリケーションが受け取ることができるテキストデータです。ユーザー発話に対して PALRO が答える処理に使うなど、自由に使用することができます。

⑤ 回答付加情報

上記③で指定したユーザー発話を音声認識した際に対応するアプリケーション名と、そのアプリケーションに引き渡す情報です。以下のフォーマットに従い記載する必要があります。

文字列の先頭に、/A=アプリケーション名/ と記載してくだい。上記③で指定したユーザー発話の音声認識を行った際、ここで指定されたアプリケーションに音声認識したことを通知します。/A=アプリケーション名/ 以降は任意の文字列を記載することができます。

※ 記述例： /A=MyApp/ JAPAN

7.2.3 プライベート会話データファイル作成例

映画情報をお知らせするアプリケーション（MyApp）が以下の流れの会話を実現する場合の作成例を示します。

【発話の流れ】

ユーザーの発話	PALRO の発話
① パルロ、上映中映画教えて	わかりました！どの国の映画がお好みですか？
②-1 日本／日本映画／邦画 （日本の映画を回答する場合）	日本ですね。しばらくお待ちください。
②-2 アメリカ／洋画 （アメリカ映画を回答する場合）	アメリカですね。しばらくお待ちください。
	現在上映している映画は、・・・・・

『パルロ、上映中映画教えて』は、アプリケーションを起動するためのトリガーワードのため、ここでは説明を省きます。上記例の②-1 は、日本の映画を教えてほしい場合に、ユーザーが回答する言葉です。この例では、『日本』『日本映画』『邦画』の3種類を想定しています。②-2 は、アメリカの映画を教えてほしい場合です。この内容を定義する例を以下に説明します。

日本[tab]ニホン[tab]ニッポン[LF]	※1
日本映画[tab]ニホンエイガ[tab]ニホンノエイガ[tab]ニッポンノエイガ[LF]	
邦画[tab]ホウガ[LF]	
アメリカ[tab]アメリカ[LF]	
洋画[tab]ヨウガ[LF]	
[LF]	※2
映画情報[LF]	※3
日本[tab]日本映画[tab]邦画[LF]	※4
日本ですね。しばらくお待ちください。[tab]/A=MyApp/ JAPAN [LF]	※5
アメリカ[tab]洋画[LF]	※6
アメリカですね。しばらくお待ちください。[tab]/A=MyApp/ USA [LF]	※7

※1） ユーザー辞書を登録します。この例では、ユーザー発話である「日本」に対し読みを2通り定義しています。これは、指定した2通りの読みのうち、どれを発話しても認識させる指定になります。また、「日本映画」に対しても同様に読みを3通り定義しています。このように、ひとつの文の読みを1行で複数指定できます。

※2） ユーザー辞書と会話タイトルの間に1行の空行を入れてください。

※3） 会話タイトルを任意の内容で登録してください。

※4） ユーザー発話を登録します。同じ意味として解釈する文を複数指定することができます。ここで指定した文に対しては、前述のユーザー辞書に読みを登録する必要があります。

※5） 回答テキストと付加情報を1行で登録します。

※6）～※7） 記載方法は、※4）、※5）と同様で、ユーザー発話の行と回答テキスト・付加情報の行を交互に指定できます。この例では、日本や邦画を音声認識した際、アプリケーション（MyApp）は、回答文として「日本ですね。しばらくお待ちください。」を、付加情報として、「/A=MyApp/ JAPAN」を受け取ることができます。また、アメリカや洋画を音声認識した際、回答文として「アメリカですね。しばらくお待ちください。」を、付加情報として、「/A=MyApp/ USA」を受け取ることができます。

・製品仕様および本書の内容は予告なしに変更される場合があります。
・PALRO は、富士ソフト株式会社の登録商標です。その他、記載されている会社名、製品名などは、各社の商標または登録商標です。

アプリケーションの開発に関する技術的な「Q&A」ホームページ

アプリケーションの開発者様からよくあるお問い合わせと解決法に関する情報を、以下のホームページで確認できます。

http://www.palrogarden.net/tecfaq/

アプリケーションの開発に関する技術的なお問い合わせ

PALRO Garden: http://www.palrogarden.net

パルロセンター
MAIL:**palro-support@fsi.co.jp**

富士ソフト株式会社 〒231-8008 横浜市中区桜木町 1-1